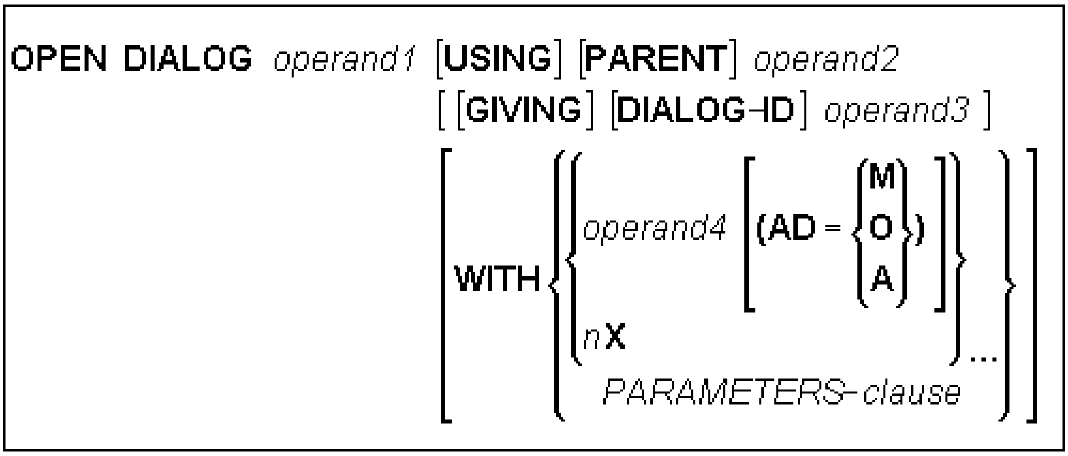
OPEN DIALOG


OPEN DIALOG operand1 [USING] [PARENT] operand2
[[GIVING] [DIALOG-ID] operand3]
WITH
operand4
(AD =
M
O
A
)
nX
...
PARAMETERS-clause

Operand	Possible Structure					Possible Formats												Referencing Permitted	Dynamic Definition
Operand1	C	S				A												yes	no
Operand2	C	S				*										G		no	no
Operand3		S							I									yes	no
Operand4	C	S	A			A	N	P	I	F	B	D	T	L	C	G	O	yes	no

* Handle

Note:
This statement is only available under Windows and Windows NT.

Related Statement: CLOSE DIALOG

Function

This statement is used to open a dialog dynamically.

Dialog Name - operand1

Operand1 is the name of the dialog to be opened.

If the PARAMETERS-clause is used, *operand1* must be a constant.

Handle Name - operand2

Operand2 is the handle name of the parent.

Dialog ID - operand3

Operand3 is a unique identifier returned from the creation of the dialog. It must be defined with format/length I4.

AD=

If operand4 is a variable, you can mark it in one of the following ways:

AD=O	non-modifiable
AD=M	modifiable
AD=A	input only

The default setting for AD= is AD=M.

Operand4 cannot be explicitly specified if operand4 is a constant. AD=O always applies to constants.

AD=M

By default, the passed value of a parameter can be changed in the dialog and the changed value passed back to the invoking object, where it overwrites the original value.

Exception: For a field defined with BY VALUE in the dialog’s parameter data area, no value is passed back.

AD=O

If you mark a parameter with AD=O, the passed value can be changed in the dialog, but the changed value cannot be passed back to the invoking object; that is, the field in the invoking object retains its original value.

Note:
Internally, AD=O is processed in the same way as BY VALUE (see the section parameter-data-definition in the description of the DEFINE DATA statement).

AD=A

If you mark a parameter with AD=A, its value will not be passed to the dialog, but it will receive a value from the dialog. AD=A fields will be reset to empty before the dialog is opened.

For a field defined with BY VALUE in the dialog’s parameter data area, the invoking object cannot receive a value. In this case, AD=A only causes the field to be reset to empty before the dialog is invoked.

Passing Parameters to the Dialog

When a dialog is opened, parameters may be passed to this dialog.

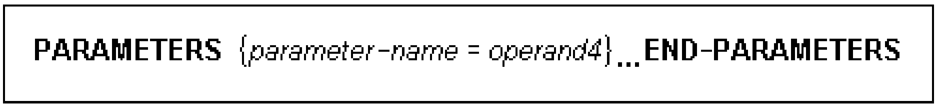
As *operand4* you specify the parameters which are passed to the dialog.

With the *PARAMETERS-clause*, parameters may be passed selectively.

*n*X

With the notation *n*X you can specify that the next *n* parameters are to be skipped (for example, 1X to skip the next parameter, or 3X to skip the next three parameters); this means that for the next *n* parameters no values are passed to the dialog.
A parameter that is to be skipped must be defined with the keyword OPTIONAL in the dialog’s DEFINE DATA PARAMETER statement. OPTIONAL means that a value can - but need not - be passed from the invoking object to such a parameter.

PARAMETERS-clause



PARAMETERS {*parameter-name = operand4*}... **END-PARAMETERS**

Note:
You can only use the PARAMETERS-clause if operand1 is a constant and the dialog is cataloged.

Parameter-name is the name of the parameter as defined in the parameter data area section of the dialog.

Note:
If the value of a parameter marked with AD=O and passed "by reference" is changed in a dialog, this will lead to a runtime error.

Further Information and Examples

See the section Event-Driven Programming Techniques in the Natural User’s Guide for Windows.